

The New Computing Infrastructure

In this issue

Things you should know.....	1
Consultants' Corner.....	2
Software and Tool News.....	4
Machine News.....	7
HPC-Behind the Scenes.....	9
Quarterly Stats.....	16
Current Machines- a snapshot in time.....	17

Things you should know...

Standard service features:

The Integrated Computing Network (ICN) Consulting Team provides user support on a wide variety of HPC topics:

- Programming, languages, debugging
- Parallel computing
- HPC and Unix operating systems, utilities, libraries
- Unix / Linux scripting
- Archival storage: High Performance Storage System (HPSS), General Parallel File System (GPFS)
- Desktop backup (TSM), storage, file transfer, network
- HPC infrastructure in both the Open and Secure

Service hours:

Monday through Friday, 8 a.m. - 12 p.m. and 1 p.m. - 5 p.m.

After-hours/Urgent support can transfer to 7x24 operations desk

Phone #505-665-4444 opt 3

Email: consult@lanl.gov

Documentation: <http://hpc.lanl.gov>

HPC Change Control Calendar: <http://ccp.lanl.gov>



Consultants' Corner

Turquoise File Transfers to the Outside

On April 2, 2015, we announced a new service for transferring data files in-between the LANL Turquoise Network and external data centers. This was the result of a joint project between HPC-3, HPC-5, CCS-7, NIE, and ADTSC. We successfully built and configured our new Data Transfer Nodes (DTNs) for high speed data movement between LANL and other Globus Online sites. Current approved transfer sites include NCAR, NCSA, NERSC, ANL ALCF, and ORNL OLCF, but you can request other endpoint connections.

For instructions, description, and DTN usage, see: <http://hpc.lanl.gov/dtn>

Request Turquoise DTN Account:

<http://hpc.lanl.gov/dtn#account>

Rules of Use:

http://hpc.lanl.gov/files/dtn_ROU.pdf

Request additional transfer site:

<http://hpc.lanl.gov/dtn#addsite>

Globus is a project of the Computation Institute, which is a partnership between The University of Chicago and Argonne National Laboratory:

<https://www.globus.org>

Transition to Lustre Scratch File Systems

As you have probably noticed, our recent expansion of scratch space on LANL HPC clusters is based on Lustre technology. In time, we will decommission all of the older Panasas-based scratch file systems.

It is straightforward to have your application move from Panasas to Lustre, simply change the pathname within the OPEN statement of your source code. However, if your application spends enough time reading and writing data files, it can likely boost bandwidth with a little extra programming effort.

The Lustre filesystems offer a convenient, easy-to-use method for adjusting file striping, and you can align your I/O calls to exploit the cluster architecture. Each Lustre filesystem is configured slightly differently with Object Storage Targets (OSTs) and our clusters have varying numbers of I/O nodes within their respective interconnect networks.

As we gain experience with Lustre at LANL, we will augment our published set of reference material with more customized suggestions for programmers on our HPC platforms:

http://hpc.lanl.gov/lustre_ref.

User Contributed Module Files

Many user applications require commonly available libraries, tools, and packages from external providers. Application developers frequently build and install them on our HPC clusters for use by their project teams. Occasionally this means replication, redundant copies of the same software built by different developers.

Several of our LANL users got together to create a place for sharing their work with the rest of the HPC community. They utilize the modulefile packaging scheme under the name: user_contrib. Now, on all of our clusters, users can load the user_contrib modulefile and view a growing list of available installed software. It is owned and managed voluntarily for users, by users.



If you would like to make a software contribution to share with other LANL scientists, send them a note: user_contrib@lanl.gov. Take a look sometime to see the list of available software, below is an example.

```
cj-fey> module load user_contrib
cj-fey> ( module avail ) | & grep -A 10 \/user_contrib
----- /usr/projects/packages/user_contrib/modulefiles/conejo -----

ParMetis/4.0.3          ensight/10.1          quo/1.2.3
SuperLU_DIST/3.3(default) global/6.3.4          random123/1.08
SuperLU_DIST/4.0        idutils/4.6           svn/1.8.10
boost/1.50.0            lapack/3.4.1          trilinos/11.10.2(default)
boost/1.57.0            lapack/3.5.0          trilinos/12.0.1
boost/1.58.0            ninja/1.4.0           visit/2.8.1
cmake/3.2.2             numdiff/5.8.1         visit/2.8.2
cmake/3.2.3             qt/5.3                visit/2.9.0
ctags/5.8               qt/5.5
cj-fey>
```



Consultants

left to right, back to front

*Rob Cunningham, Ben Santos, Michael Coyne,
Rob Derrick, Giovanni Cone*

*Front Row: Peter Lamborn, David Kratzer, Hal
Marshall*

Software and Tool News

The Programming & Runtime Environments Team strives to provide timely, reliable, and robust libraries, compilers and tools for LANL's HPC customers. Additionally, experts are brought in to deliver free workshops for all LANL badge-holders. Stay tuned to the ICN notification system emails for these announcements, as these sessions offer unique opportunities to gain insight into new features and one-on-one assistance with tough programming obstacles from the developers of the performance and debugging tools we provide.

The TAU Performance System

The Programming and Runtime Environments (PRE) Team recently hosted Dr. Sameer Shende of the University of Oregon. During Dr. Shende's two-day visit to LANL, he provided a workshop for interested individuals and personalized consultations for users who are currently using the Tuning and Analysis Utilities (TAU) toolset. Workshop participants learned through hands-on examples that TAU's aim is to assist with the investigation of the behavior and performance of parallel programs. Workshop participants used the latest version of TAU, version 2.24.1, available on all turquoise and yellow HPC systems. Our installations are interoperable with your favorite compiler and MPI library, featuring many bug-fixes

over prior releases. Simply ``module load`` your compiler, MPI library and the TAU module file (in that order) to enable the TAU software in your Linux environment.

TAU provides three levels of program integration, and each one offers the capability to capture helpful performance data of an application as it runs on HPC systems. At the first level, TAU can track and provide event based sampling data of MPI, I/O, and OpenMP calls, simply by using some execution flags in conjunction with the "mpirun" command. At the second level, TAU has the ability to rewrite compiled binaries to obtain data about program function calls. This level also has the ability to query available hardware-counter information, such as data cache misses, to help characterize utilization of computing resources. At the most intensive level, source code instrumentation, a finer granularity of performance can be acquired, such as characterization of loops and a program's level and efficacy of vectorization.

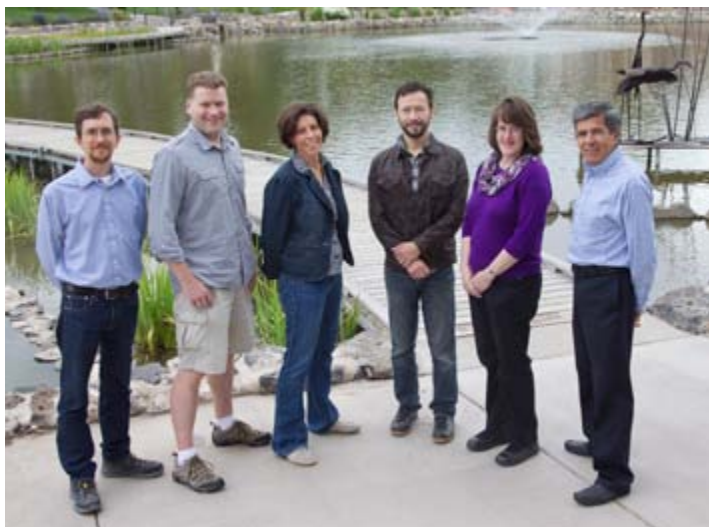
Performance data generated by running a program under TAU can be visualized using several useful graphical analysis tools. ParaProf comes with TAU and provides detailed analysis of profiling data, including 3-D topologies. Tracing results can be visualized with Vampir, a powerful, licensed visualization tool made available by a separate environment module on LANL's HPC systems.

Thanks to all who participated! If you missed this workshop and would like assistance using this ultra powerful performance analysis utility, let the team know and we'll assist you in getting started: ptools_team@lanl.gov. More information about TAU can be found at <http://tau.uoregon.edu>.

Programming and Runtime Environments Team

left to right

*Sean Brennan, David Shrader, Jennifer Green,
Giovanni Cone, Monique Morin,
and Jorge Roman*



Paraview Catalyst Deployed on LANL HPC Platforms

With assistance from Paraview's vendor, Kitware, the PRE team is proud to announce that Paraview is now available for use in Client-Server visualization sessions and for use with applications using Paraview Catalyst's in-situ visualization capability. More information on using the HPC provided Paraview in a Client-Server session is found on the following webpages:

For Yellow and Turquoise TLCC platforms: http://hpc.lanl.gov/paraview_usage_yeti

For Secure TLCC platforms: http://hpc.lanl.gov/paraview_usage_red

Note: Deployment of an HPC supported Paraview on Cray platforms is currently underway.

Intel VTune Amplifier Hardware Counter Access

Optimization and vectorizing are crucial for best performance of computer codes in newer processors. As such, Intel performance analysis tools, such as VTune Amplifier, can be useful. VTune version 15 update 3 is available on all systems. Intel's Sampling Driver kernel modules for performance analysis have been installed on Pinto's compute nodes to provide access to hardware counters related to memory access. Therefore, Pinto Compute Nodes can be used to leverage complete VTune functionality. Other systems may get the kernel modules installed after performance impact and usage is assessed on Pinto.

VTune Graphical User Interface allows complex analysis without the need to instrument the code. Module load intel-amplifier and then invoke amplxe-gui to get started. A tutorial for VTune can be found at:

https://software.intel.com/sites/default/files/managed/c0/a8/hotspots_amplxe_lin.pdf

Allinea Forge in Client-Server Mode

Users of Allinea's DDT debugger and MAP profiler may notice a significant lag when running these tools over a remote X11 tunnel. This can become a painful experience. That pain is no longer necessary. Thanks to Allinea's remote launch mechanism and the LANL specific customization of the installations on HPC resources, Linux and OSX users can now configure Allinea client software to run locally on a desktop as their parallel application runs on a cluster. By leveraging the parallelism and scalability of the cluster installation, as well as the Allinea site license, customers can more easily run this graphical tool using the local desktop's graphical capabilities via a client-server connection.


Navigate to http://hpc.lanl.gov/allinea_ddt for full site-specific instructions.

Need assistance? Email ptools_team@lanl.gov

The Programming & Runtime Environments Team is an important resource available to LANL developers. We aim to provide a user-friendly, yet robust programming environment upon which our lab's scientific excellence can be upheld. If ever you have any questions or require any tool or implementation specific assistance upon our clusters, please reach out to us. We will work hard to help you overcome software obstacles so that you can focus on getting results on our High Performance Computing systems.

How to Prepare for Trinity, and Beyond

As the Trinity platform begins to come into focus, code developers are asking, "How do I prepare my code for it?" Optimizing production codes often requires refactoring, structuring and redesigning algorithms befitting of the target system's architecture. This is a difficult task as the changes made today should be in preparation for anticipated future systems.



A big advantage to Trinity over heterogeneous systems is that special accelerator off-loading is not required to use the Haswell / KNL architecture. This is not to say that MPI everywhere is the suggested approach. In order to maximize efficiency on Trinity, multiple levels of parallelism, including vectorizing ideal code regions, is key. Not all codes are well-suited for this, but there are ways to gain insight into how code is running on resources, make use of tools to tune for maximum performance, and have that effort pay off on Trinity and other platforms. This is an exercise all code developers should be considering. According to a CERN 2013 study (Nowak, Andrzej. "Software Optimization for the Many-Core Era - The CERN Case." <http://openlab.web.cern.ch/sites/openlab.web.cern.ch/files/presentations> CERN, 13 Mar. 2013. Web. 3), tuning an application during the development process is crucial.

In the study, the primary tasks delineated include:

- Algorithm tuning offers several orders of magnitude performance gain
- Source code optimization can provide up to an order of magnitude speed-up
- Compiler optimization provides at least 10% to 20% improvement, potentially more

Restructuring and tuning the application's algorithm proves most beneficial. Basic decomposition of the problem, aligning its solution to reduce data movement, maximize asynchronous computation, improving memory access, aligning check-pointing to what makes sense for the reliability of the system, intelligent file access patterns, and exploiting multiple levels of parallelism available are all important to consider in algorithm optimization. While it's quite a tall order, an algorithm adjustment or redesign may be required to achieve good performance in HPC.

If you already have fine-tuned the algorithm in established source-code, software tools can help you analyze your application to identify

undetected bottlenecks or imbalances. The Programming & Runtime Environments Team provides installations of licensed and open-source third-party performance analysis tools on all LANL HPC production clusters. Some examples are Gprof, VTune Amplifier, AllineaMAP, TAU, Open|Speedshop, HPCToolkit, ScoreP and Perftools. These tools identify where your current code may be improved. Some are lightweight and give high-level performance indicators through dynamic instrumentation of compiled code via sampling algorithms or autonomous resource monitoring techniques. Others are heavy duty and capture application performance through manual or automatic source code instrumentation mechanisms. The features vary widely per tool and each unique use-case generally dictates the appropriate tool to use. LANL's ICN Consulting Office and Programming and Runtime Environments Team are on hand to help you start analyzing the performance of your application. We encourage ongoing use of performance analysis tools in the software development workflow so bottlenecks and imbalances can be addressed early-on.

Compiler optimization features offer performance improvements. The flags provided to the compiler control this. This last step is one where most mainly rely on the compiler to implement the optimization, since programming intrinsics are architecture specific. Directives (pragmas) can guide the compiler, but it is up to the compiler to execute this step correctly. Verifying the compiled code after making use of their optimization flags is the responsibility of the developer, and is always advisable.

Using a different compiler can affect the results, and even a newer version of the same compiler can introduce differences. The optimization flags sometimes trade-off accuracy for speed. In some applications this may be acceptable, but in others it may not be. Therefore, verification of your

results should be a recurring part of the software development lifecycle. Changes to anything from the runtime environment up to the optimization flags could change your results. To drive this point home, an application may underutilize Trinity's Knights Landing (KNL) compute capability if vectorization is not implemented. The report, "Vectorization of ChaCha Stream Cipher," (Goll, Martin, and Gueron, Shay. "Vectorization on ChaCha Stream Cipher." 11th International Conference on Information Technology: New Generations (ITNG), 2014. IEEE, 2014.), estimates that vectorizing can reduce runtime by 50% for AVX256 processors (Haswell) and speculates that it could be reduced an additional 50% with KNL's AVX512. In other words, if a code does not vectorize, it may only be able to use less than half of the KNL processor capabilities. Future processors are expanding their vector register lengths, making vectorization necessary to achieve peak performance.

Writing compiler-agnostic code will permit you to try out the various compilers in order to see which one will optimize your code best. One size does not fit all in this case, so portability is strongly recommended. Finally, note that any optimization tuning you do on your code now will offer better performance on future processor architectures.

Machine News Future Infrastructure

Paul M. Weber, HPC-5

Planning, provisioning, and preparing for future compute platforms is a continuous process. Even though phase one of Trinity has just arrived at our facility, we have already begun the gathering and planning process for our next round of infrastructure upgrades to accommodate the ATS-3 (Crossroads) and ATS-5 platforms. These are slated to arrive in 2020 and 2025, respectively.

The technological foundation for HPC systems is no less volatile and subject to disruption than is that of the cellular telephone, personal computing, or other such electronics industry. Though contemplating the capabilities, form factor, and other such features of the newest Apple® gadget-10 years from now-may seem like a speculative and fanciful exercise, this is very akin to the effort we must put forth in planning for our next generation infrastructure.



Installing cooling system under Trinity



Piping under Trinity

To make planning tractable, HPC division is continuously working to stay abreast of development trends, potential disruptions, vendor initiatives, and technological limitations in the HPC arena. Much benefit is gained from our relationships with other DOE computing centers. Additional insight is regularly gleaned directly from vendors, either through non-disclosure agreements, or via working relationships associated with our system acquisition contracts. A number of DOE funded initiatives, including Non-recoverable Engineering (NRE, associated with ATS system acquisitions), Fast Forward, and Design Forward, all allow DOE laboratories a unique opportunity to participate in vendor design efforts that lead the way to exascale technology. Ultimately, however, projecting the power, cooling, floor loading, and other infrastructure requirements of 2025 is largely a subjective exercise. We must marry these various fragments of information from industry with LANL's rich historical knowledge of growth trends in HPC system capabilities and requirements.

Broadly, what can be seen from current trends in the chip industry is that geometric scaling (Moore's Law) is beginning to wane due to impending limitations in semiconductor lithography technology. Though manufacturers are continuing to obtain advances in performance, by turning to other enhancements in chip design, improvements in power efficiency are traditionally a by-product of geometric scaling. Increasing the performance or capability of a processor, without an appropriate reduction in transistor size generally implies a greater demand for power. This is driving chip designers to craft other clever means for improving power utilization, but current projections indicate that the cost per flop of chip performance will generally outpace power efficiency in the years ahead. The bottom line is that future platforms will continue to place greater power and cooling demands on our facilities.

These projections are heavily clouded by national efforts to accelerate the arrival of exascale computing. Continuing changes and funding fluctuations within the national Exascale Computing Initiative (ECI) impose uncertainty in the requirements and the delivery schedule for exascale systems. There is further uncertainty in how the ASC program within NNSA will ultimately participate in, and benefit from, these efforts. Without a revolution in compute power efficiency, the delivery of an exascale system to LANL in the 2025 time frame has substantial implications for our infrastructure requirements.

With the recent upgrade project, retrofitting the piping, pumping, and cooling infrastructure for direct warm-water cooling, the originally designed power and cooling capacity of the SCC has already been well exceeded. The newly installed cooling loop has a designed operational capacity of approximately 34MW, once it is fully outfitted with the necessary towers and heat exchangers. With some internal configuration changes, power to the computer floor will soon be expanded to 24MW. Subsequent modifications necessary to bring an additional 10MW to the floor are currently being studied. Barring an early acquisition of an exascale class system, we still have a good deal of wiggle room in the operational capacity of our facility. Each of these expansion efforts is a costly



Water cooling system pumps

endeavor. With each expansion, a new degree of complexity is introduced to facility operations. Though the efficiency of warm-water cooling is substantially superior to air cooling, it necessitates greater initial investment to tie the platform into the cooling infrastructure. With such a tremendous load placed on the electrical grid, greater sophistication will be required to adaptively manage facility power usage. Further, our facilities are aging. Original equipment installed in the SCC in 2002-3 is beginning to reach end of life. All in all, it is clear that costs associated with power, cooling, and facility operations (along with storage, networking, and other peripheral systems) are now outpacing core compute costs. As we continue to press the limits of computational capability, we will certainly be pressing the limits of the infrastructure that supports it.



Electrical Switchboard

HPC-Behind the Scenes



**Strategic Computing Complex (SCC)
Computer Cooling Equipment Project**
Phil Sena, HPC-DO

The Advanced Simulation and Computing (ASC) Program is vital to LANL in its role as part of NNSA's collaborative program among Lawrence Livermore, Los Alamos and Sandia national laboratories, ensuring the safety and reliability of the nation's nuclear weapons stockpile. Supercomputing platforms such as Roadrunner, Cielo and Trinity are well known supercomputing platform projects of the ASC Program.

It is not well known that in preparation for future phases of supercomputing at LANL's High Performance Computing facilities, major mechanical and electrical equipment installation are required in order to provide the necessary cooling and power capability for ASC Platforms. The Nicholas Metropolis Center for Computing and Simulation(SCC), is home to these major infrastructure equipment projects. These projects are conceptually under design multiple years ahead of future platform deliveries to stay ahead of the power and cooling needs these future platforms demand.

The SCC Computer Cooling Equipment (SCC-CCE) Project was recently completed in May 2015 in time for the arrival of the ASC Program's Trinity supercomputing platform. All previous cooling

equipment projects were based on cooling computer platforms with “air”, using large amounts of power based on chilled water (mid 50s degrees) processing and large amounts of groundwater necessary for cycling through cooling towers outside the facility. This water cooling capability provided by the project is expected to help service the needs of multiple future machines.

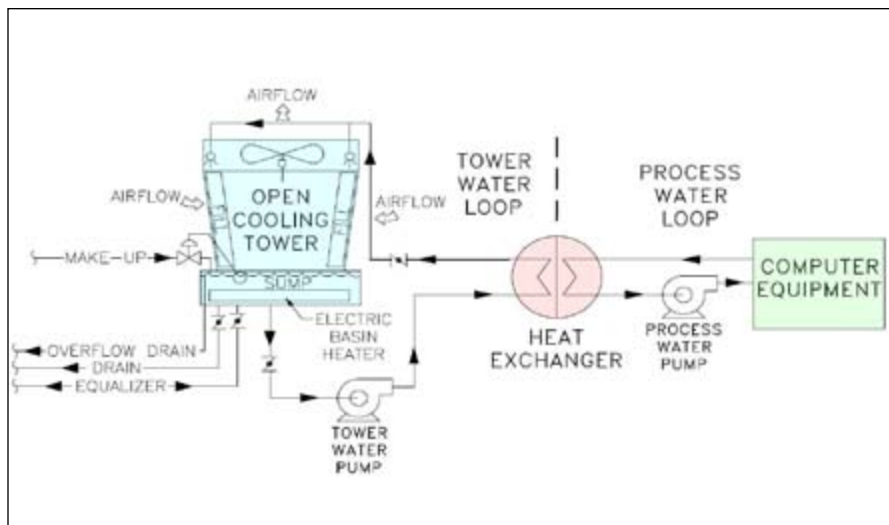


New SCC Cooling Towers

The SCC-CCE Project adopted the concept of warm water cooling technology, a new concept to LANL. The technology involves the use of warmer temperature water (approximately 70 degrees) transported directly from new cooling towers to heat exchangers instead of the previous air cooled concept using high power consuming chillers. This will result in projected large-scale energy savings that will be detailed in subsequent articles in this newsletter. A closed loop of processed water furnishes computer platform racks with this warmer water, heating it to approximately 85 degrees, returning it to the heat exchangers for return and re-cooling to the cooling towers. During the project construction period the Sanitary Effluent Reclamation Facility (SERF) was completed and became operational. SERF will furnish up to 88 million gallons of recycled water to the SCC for cooling tower use and will directly reduce the amount of total groundwater previously used for air cooled computer technology.

Once the platform technology was finalized to connect Trinity to the new warm water cooling capability provided by the SCC-CCE Project, the Physical Infrastructure Integration (PII) for Trinity Project was required to design and install the process cooling system. The process water system, furnishing warm water-cooling directly to the platform’s racks involve ten (10) individual eight (8”) inch diameter-cooling loops under the SCC computer floor. The loops circulate cooling water constantly in order to meet the cooling and pressure capability required for the platform racks. The process system temperatures and pressures are controlled and monitored using the facility’s automated controls monitoring system.

From a power standpoint the SCC-CCE project consists of basic power requirements to control the many pumps, automated control valves, heat exchangers, cooling towers and miscellaneous support equipment. The major requirement for power is the actual computer platform system racks on the computer floor. In order to provide almost ten (10) Megawatts of power to the computer floor the SCC-CCE Project procured two (2) large electrical substations. Upon completion of the SCC-CCE project the HPC-2 electrical facilities team installed the final routing of power cabling from new switchboards to the computing systems. The SCC-CCE Project is the first in a series of cooling equipment projects in preparation for



SCC-CCE Warm Water Cooling Concept

a series of Advanced Technology Systems (ATS) computing technology systems to be located at LANL. At the time of design inception, the ASC Program provided the best power and cooling projections through 2025 available and these were incorporated into the final design of the SCC-CCE Project. Future ATS platforms are expected in 2020 and 2025 and other computing projects for the ASC program are being planned. These future demands on the SCC physical infrastructure are driving requirements for the next SCC power and cooling project to be initiated soon.

LDCC Cooling Infrastructure

Cindy Martin, HPC-2



The Laboratory Data Communication Complex, or LDCC, built in 1989, includes 20,000 square feet of data center space and primarily houses our Open Science and ASC testing platforms. It is capable of providing 8MW of electrical capacity, 9MW of central air cooling capacity, and with recent enhancements, 2MW of water cooling capacity to the HPC data center.

HPC anticipated the increase in power density of High Performance Computing (HPC) components would necessitate the use of water based solutions for heat transport

rather than traditional air cooling solutions. As a result, facility infrastructure projects were initiated to address this need in 2011 with the Petaflop project. The Petaflop project included adding the heat exchangers, process cooling pumps, and process and return water loops to expand the LDCC’s cooling capabilities. This infrastructure was the first step in adding 2 MW of HPC cooling capacity to the data center floor. The diagram (below) details that infrastructure.

In 2014, with the ACES Cray XC40 testing platforms, infrastructure to distribute water to the data center floor was completed. This required the addition of water process loops as well as feedback mechanisms to maintain proper water temperature, pressure, and flow rates. The water process loop is depicted in the two diagrams at the end of this article.

Trinitite and Gadget, the new Cray XC40 test beds, were connected to this loop in February 2015. A primary focus of these test beds includes the collection and correlation of environmental data from the platform and building automation systems.

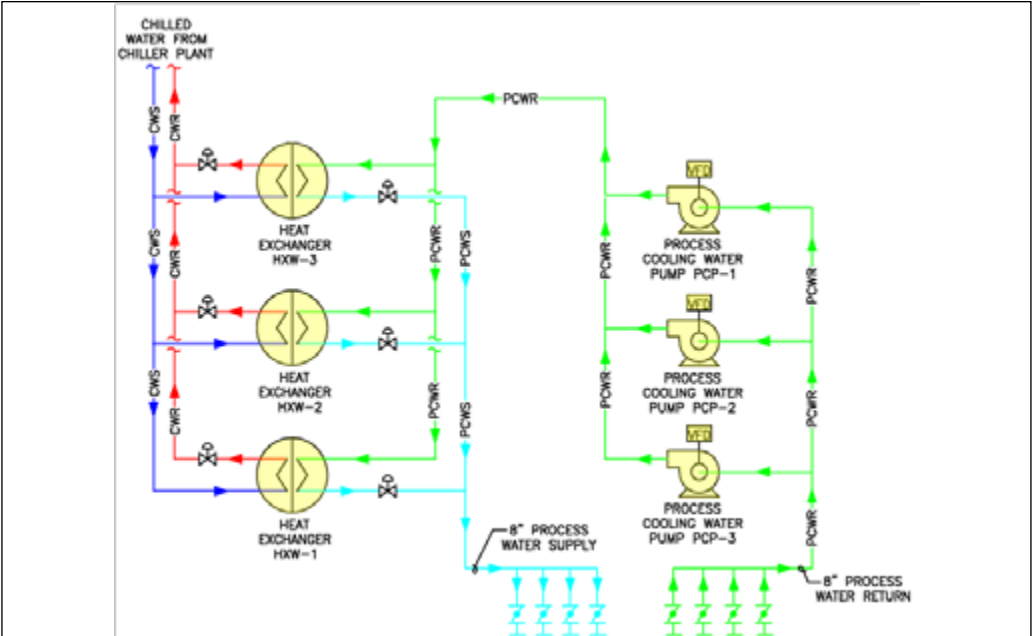
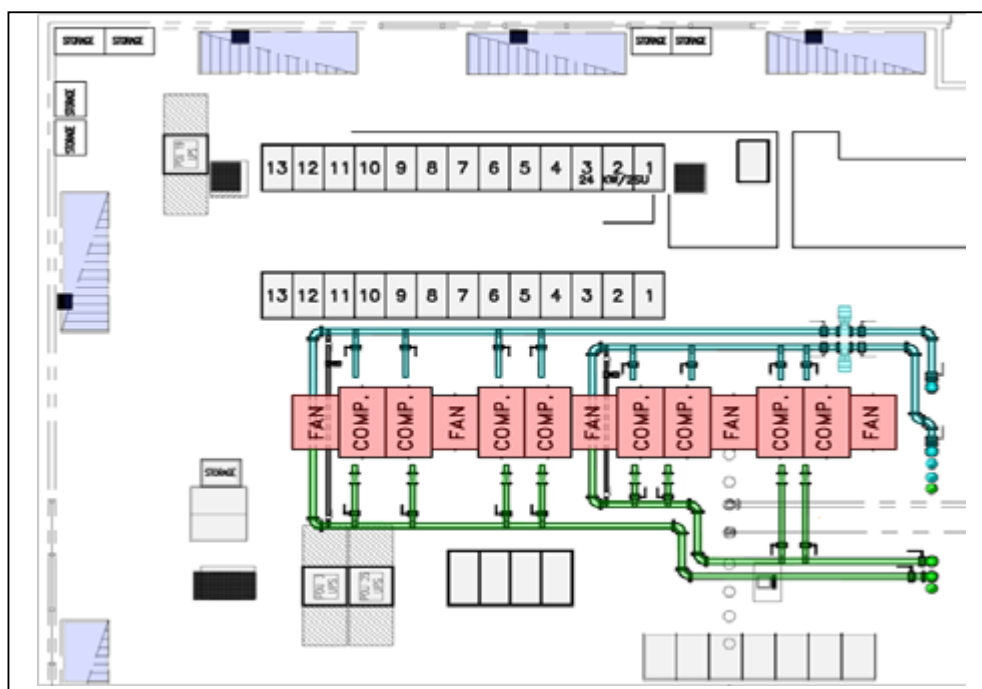
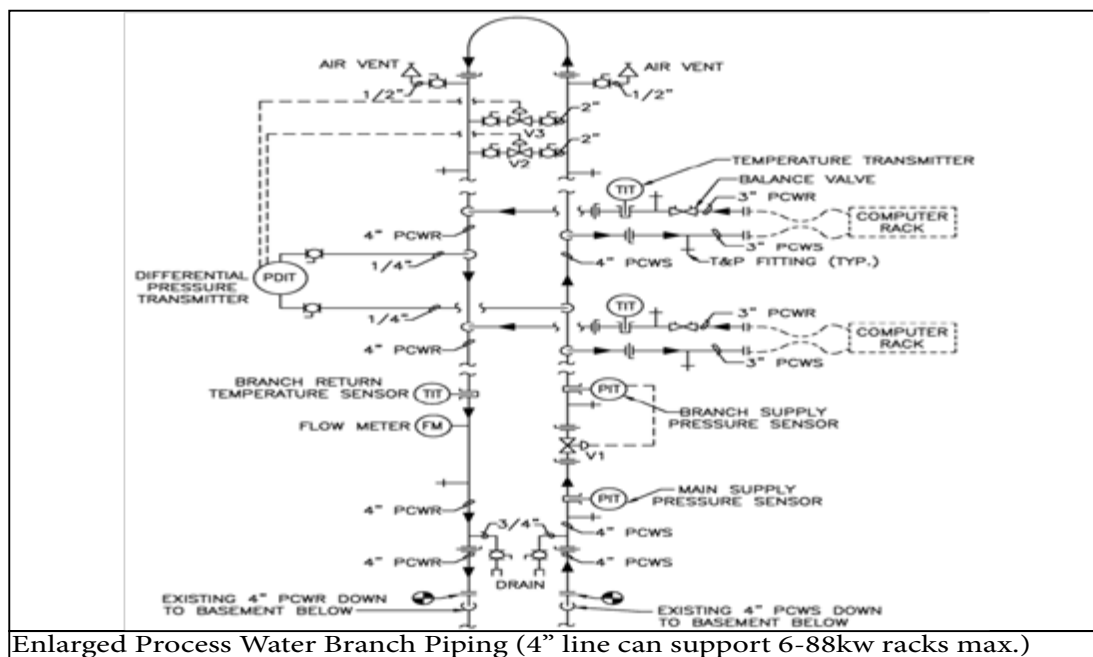


Diagram of LDCC’s cooling infrastructure

With Trinitite, the ACES facilities and monitoring teams have been able to test some of the power management capabilities of the Cray and correlate those to facility power draws.

HPC platforms are now pushing the limits of data center power and cooling infrastructure. In order to maximize the value of modern large scale platforms a management approach that

tightly integrates all information, both internal and external to the platforms, is required. The ability to control data center infrastructure dynamically based on platform, job, power, and environmental information will become a necessity as we move towards exascale computing.



Sanitary Effluent Reclamation Facility (SERF)



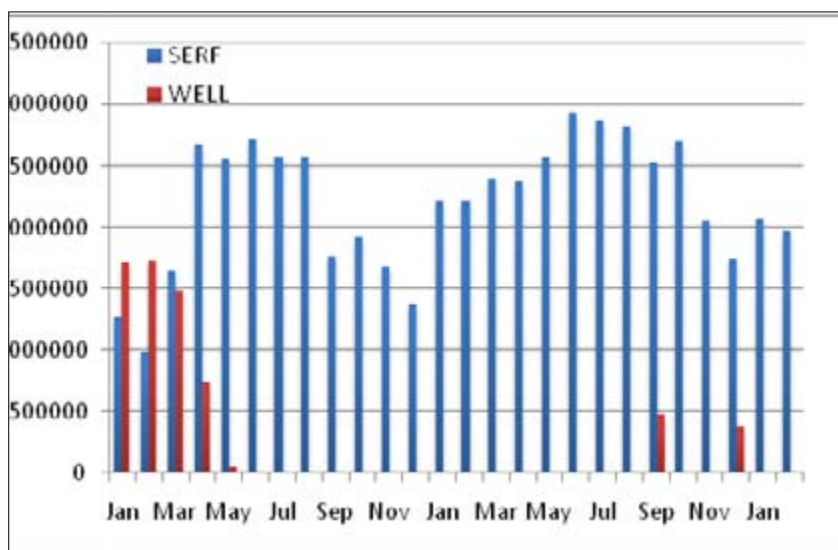
Originally built in 2003 to test technology for treating sanitary effluent, this 4,350-square-foot facility is located east of the TA-3 Steam Plant. The Sanitary Effluent Reclamation Facility (SERF) cleans sanitary effluent to a standard better than drinking water. The SERF treats effluent from the sanitary wastewater treatment plant. It includes storage tanks for 1,500 gallons of hydrochloric acid solution, 4,500 gallons of sodium hydroxide solution, 500 gallons of ferric chloride, and 4,500 gallons of magnesium chloride solution.

This facility allows the Lab to process up to 100 gallons per minute of sanitary effluent or 120,000 gallons of water a day. The reclamation facility contributed more than 27 million gallons of re-purposed water to the Strategic Computing Complex (SCC). Using treated sanitary effluent in the cooling towers reduces LANL's use of potable water for such purposes, helping to meet a Department of Energy/National Nuclear Security Administration requirement for recycling and reducing water use.

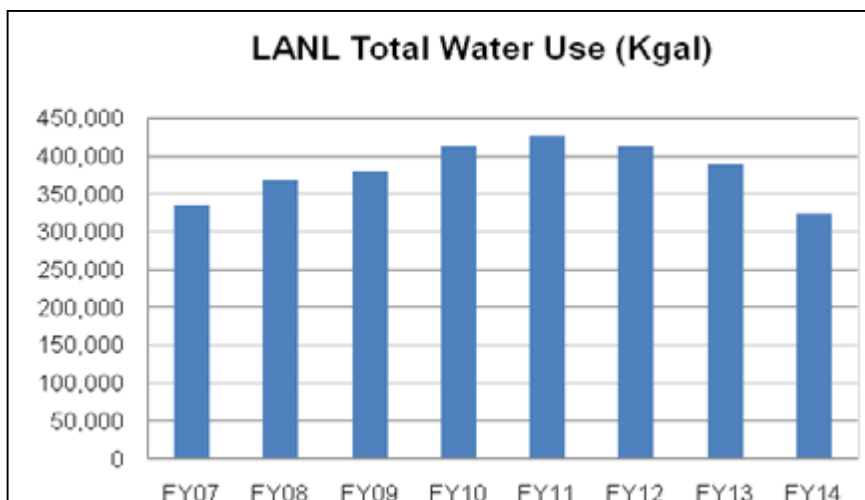
LANL decreased its water usage by 26 percent in 2014, with about one-third of the reduction attributable to using reclaimed water to cool our supercomputing center. The SERF plant provides all the water necessary to operate the cooling infrastructure at the

SCC. The SERF is also capable of supplying the water needed for the new Trinity platform.

"Our goal during 2014 was to use only re-purposed water to support our mission at the Strategic Computing Complex (SCC), and we achieved that goal," said Cheryl Cabbil, associate director of Nuclear and High Hazard Operations, which administers the Sanitary Effluent Reclamation Facility (SERF). "Part of our role as good stewards of the environment is to conserve finite resources such as water whenever possible," said Michael Brandt, associate director of Environment, Safety and Health at the Laboratory. Conserving water while achieving our mission is a great example of how we are pursuing long-term environmental sustainability. A [video](#) that explains how SERF works is available online.



SERF vs. Well water use in 2014



LANL total water use 2007-2014

Power Projects for Trinity

by Josip Loncaric,HPC-DO

High performance computing power consumption is currently the main driver of changes in computing technology. To illustrate the point: If we built an exascale computer out of 1,000 Roadrunner machines, it would probably be included for free with the contract for a nuclear power plant required to turn it on. This problem isn't limited to high performance computing. Exponentially increasing demand for information processing at all levels, from smart watches to cell phones to televisions and computers implies growing power demands for all electronics. When IEEE Spectrum interviewed Bernd Hoefflinger in 2012, he predicted:

"They expect 1000 times more computations per second within a decade. If we were to try to accomplish this with today's technology, we would eat up the world's total electric power within five years. Total electric power!"

Clearly, exponential growth in computing without corresponding growth in the world's electricity generating capacity is possible only with exponential growth in energy efficiency of computing. The entire semiconductor industry is laser-focused on this problem. What makes this hard is that Dennard scaling for CMOS semiconductors has already stalled. A brief technology refresher is in order.

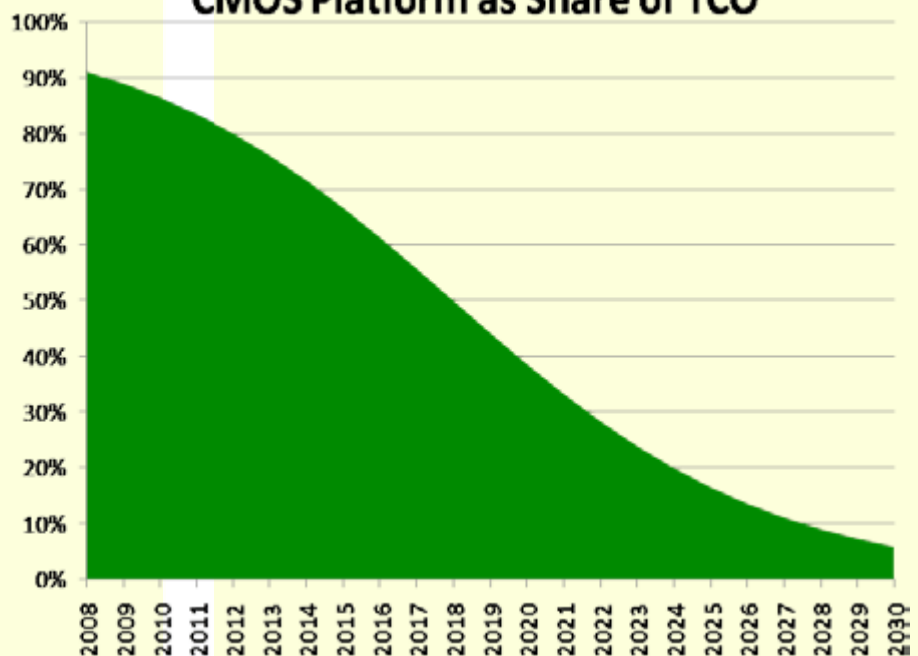
CMOS technology has become ubiquitous in part because it is extremely power efficient: ideally, power flows only when voltage changes. How much power? The amount of energy required to flip a bit is proportional to the capacitance, which scales with lithography feature size. When geometric scaling slows

down due to lithography challenges, so do power efficiency gains.

So, what's the alternative? Semiconductor manufacturers have been pursuing two main initiatives: lower the operating voltage, and apply aggressive power management. Voltages can be lowered, but only down to near silicon threshold voltage, where reliable operation requires decreased operating frequencies, which in turn necessitates more transistors to compensate for the performance loss. The upside potential of this approach has been estimated to be at most 9.6x higher power efficiency.

Aggressive power management is a different strategy which quickly turns off anything that isn't in use. Sometimes, this strategy is called "run to idle" where idle power is reduced. This has a substantial promise in applications where only a fraction of the resources are in use at any one time. For example, if a program can utilize at most 10% of the chip's capabilities, nearly 90% can be turned off while the useful 10% continues under full power.

CMOS Platform as Share of TCO



This of course entails complex power management strategies, already built into today's processors, which typically results in performance variability. This has dramatic implications for today's parallel programs, which are bulk-synchronous and expect processors to behave in lockstep, like a chorus line. Modern processors are different, they behave more and more as free-style jazz dancers where each one manages its own performance. This means that the slowest processor governs the progress rate of the whole parallel application. This reason, as well as demands of dynamic load balancing, are forcing a new look at task parallel programming models which relax synchronization requirements between tasks.

Moreover, large scale parallel applications are capable of multi-MW power transients in milliseconds, as applications transition from a low power phase (such as I/O) to a high power phase (heavy computations) or vice versa. Provisioning the data center with sufficient power to allow power peaks is very expensive and quite inefficient since the highest peaks are rare.

As a result, we are interested in power capping capabilities, where platforms, jobs, and nodes manage their power consumption to stay within their power allocation. A processor can do that by slowing down. The challenge is to reliably and quickly coordinate power management of nearly 100,000 independent parts in the system, and make this control responsive to the overall objectives of the data center. Much work remains to be done in this area, before we can maximize the computational benefits possible under the variable power availability conditions. Power adaptive computing is in our future.



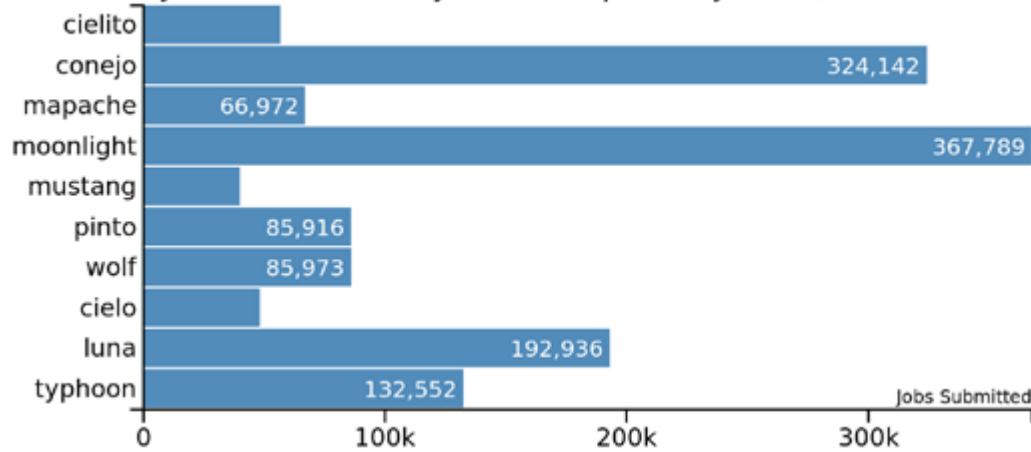
Process cooling water system in air handling room



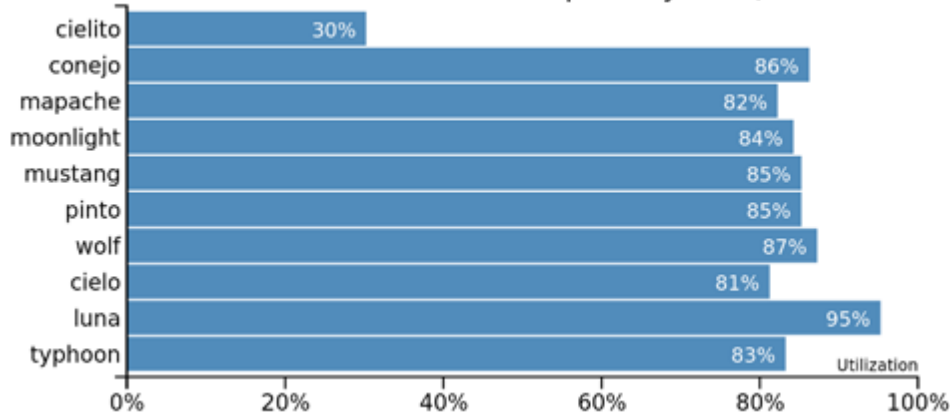
Craftsmen tighten bolts on cooling tower piping system

Quarterly Statistics

Number of Jobs Submitted by Users - Apr 1 to Jun 30, 2015

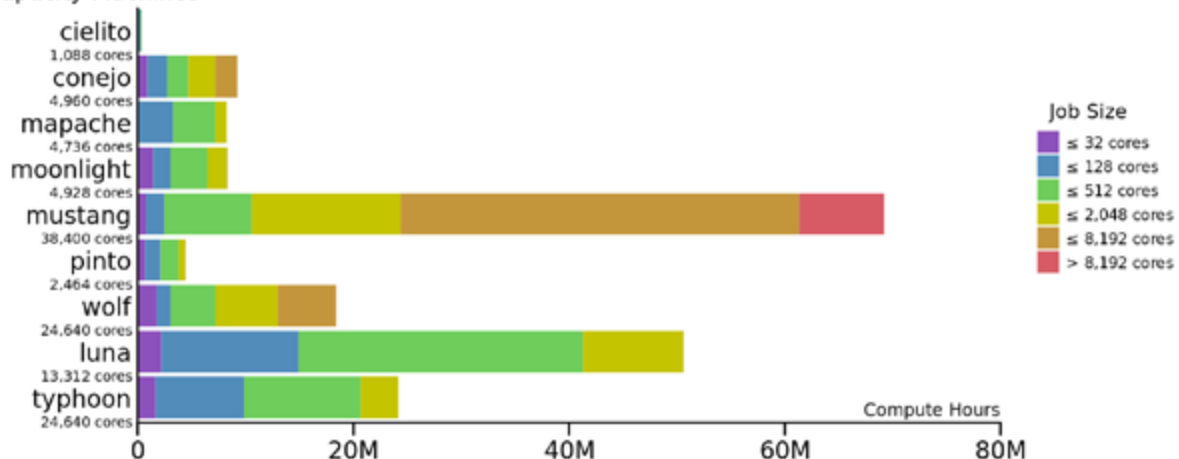


Percent of Total Possible Time Utilized - Apr 1 to Jun 30, 2015



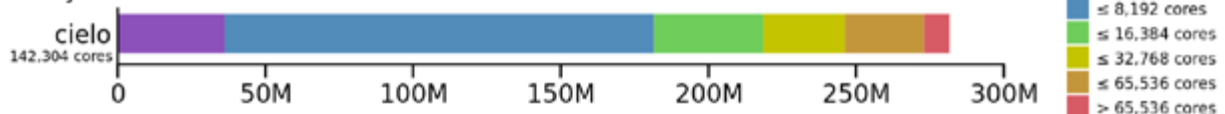
Total Compute Time of All User Jobs - Apr 1 to Jun 30, 2015

Capacity Machines



Total Compute Time of All User Jobs - Apr 1 to Jun 30, 2015

Capability Machine



Current Machines- A snapshot in time

Name (Program ¹)	Processor	OS	Total Compute Nodes	CPU cores per Node/ Total CPUs	Memory per compute Node/Total Memory	Interconnect	Peak (TFlop/s)	Storage
Secure Restricted Network (Red)								
Cielo (ASC)	AMD Magny-Cours	SLES-based CLE and CNL	8,894 nodes	16/142,304	32 GB/297 TB ⁵	3D Torus	1,370	10 PB Lustre
Luna TLCC2 (ASC)	Intel Xeon Sandybridge	Linux (Chaos)	1540 nodes	16/24,640	32 GB/49 TB	Qlogic InfiniBand Fat-Tree	513	3.7 PB Panasas
Typhoon (ASC)	AMD Magny-Cours	Linux (Chaos)	416 nodes	32/13,312	64 GB/26.6 TB	Voltaire InfiniBand Fat-Tree	106	3.7 PB Panasas
Open Collaborative Network (Turquoise)								
Cielito (ASC)	AMD Magny-Cours	SLES-based CLE and CNL	68 nodes	16/1088	32 GB/2.3 TB ⁵	3D Torus	10.4	344 TB Lustre
Conejo (IC)	Intel Xeon x5550	Linux (Chaos)	620 nodes	8/4960	24 GB/4.9 TB	Mellanox Infiniband Fat-Tree	52.8	1.8 PB Panasas
Lightshow ³ (ASC)	Intel Xeon	Linux (Chaos)	16 nodes	12/192	966 GB/1.5 TB	Mellanox Infiniband Fat-Tree	4.0	1.8 PB Panasas
Mapache (ASC)	Intel Xeon x5550	Linux (Chaos)	592 nodes	8/4736	24 GB/14.2 TB	Mellanox Infiniband Fat-Tree	50.4	1.8 PB Panasas
Moonlight TLCC2 ³ (ASC)	Intel Xeon E5-2670 + NVida Tesla M2090	Linux (Chaos)	308 nodes	16/4,928 + GPUs	32 GB/9.86 TB	Qlogic Infiniband Fat-Tree	488	1.8 PB Panasas
Mustang (IC)	AMD Opteron 6176	Linux (Chaos)	1,600 nodes	24/38,400	64 GB/102 TB	Mellanox Infiniband at-Tree	353	1.8 PB Panasas
Pinto TLCC2 ³ (IC)	Intel Xeon E5-2670	Linux (Chaos)	154 nodes	16/2464	32 GB/4.9 TB	Qlogic Infiniband Fat-Tree	51.3	1.8 PB Panasas
Wolf TLCC2 ³ (IC)	Intel Xeon E5-2670	Linux (Chaos)	616 nodes	16/9856	64 GB/39.4 TB	Qlogic Infiniband Fat-Tree	205	1.8 PB Panasas

¹ Programs: IC=Institutional Computing, ASC=Advanced Simulation and Computing, R=Recharge

³ TLCC = TriLab Linux Capacity Cluster; 2 = 2nd Generation

⁵ Cielo has 372 viz nodes with 64GB memory each

⁶ Cielito has 4 viz nodes with 64GB memory each